TX-0 COMPUTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE 39, MASSACHUSETTS

M-5001-39

DOCTOR

October 29, 1962

DOCTOR

DOCTOR (Midas Debugger) is a symbolic debugging program for the TX-0. It occupies upper memory starting at register 15000. In addition, the symbol table builds down into lower memory. The initial symbol table includes all TX-0 operations, and occupies down to register 14455. Entry point of DOCTOR is at register 15000.

Operation of DOCTOR is from the on-line Flexowriter. Lower case letters and numerals are symbol constituents; other characters, and characters typed in upper case are either control characters or are illegal. A symbol consists of one to six letters and/or numerals, of which at least one must be a letter. A string of digits alone is taken as an octal or decimal number. The symbol syntax is the same as that of MIDAS. DOCTOR will read absolute binary tapes produced by MIDAS, and also MIDAS symbol punches for the purpose of informing itself of the user's symbol definitions.

One of DOCTOR's most useful features is the "breakpoint." When debugging a program, it is occasionally desirable to allow control to flow up to a certain point, at which the programmer would like to examine the contents of the AC, LR, XR and various other registers in his program. To facilitate this, DOCTOR will insert into the user's program a transfer instruction into itself, which will cause the AC, LR, and XR to be saved and printed out. It is then possible to examine arbitrary program locations, make changes as necessary, move the breakpoint if desired, and continue the program, restoring all indicators and executing the instruction which was originally replaced by the breakpoint transfer.

The following list describes the action of typed in characters.

<table>
<tr><th>CHARACTER</th><th>ACTION</th></tr>
<tr><td>space</td><td>separation character meaning arithmetic plus.</td></tr>
<tr><td>+</td><td>separation character meaning arithmetic plus.</td></tr>
<tr><td>-</td><td>separation character meaning arithmetic minus.</td></tr>
<tr><td>|</td><td>register examination character; preceded by an address, causes the addressed register to be opened, and the location sequence to be reset to this address. Immediately following a register printout, it will cause the register addressed therein to be opened. Opening a register causes the contents to be typed out as an instruction or constant, according to the current mode and makes the contents available for modification.</td></tr>
<tr><td>(</td><td>same as |, but forces printout as octal constant for this examination.</td></tr>
<tr><td>/</td><td>same as |, but forces printout as instruction for this examination.</td></tr>
<tr><td>carriage return</td><td>if a register is open for examination and any expression has been typed immediately prior to the carriage return, the value of that expression is stored in the open register. Otherwise, no change is made.</td></tr>
<tr><td>backspace</td><td>has same effect as carriage return, but then opens the next sequential register. This sequence is not altered by additional |, (, or / characters typed after a register has been opened.</td></tr>
<tr><td>tape feed</td><td>same as backspace, but opens the previous register.</td></tr>
</table>

CHARACTER	ACTION
tab	same as carriage return, but opens the register addressed by the contents of the last opened register (after modification, if any). Tab alters the sequence of locations.
=	types out the last quantity as an octal integer.
I	types out the last quantity as an instruction.
3	types out the last quantity as flexo code, in the order right, middle, left.
.	as a single symbol, has the value of the current location. Following a string of digits, means decimal (integer).
A	has the value of the location in which the preserved <u>accumulator</u> is stored.
L	has the value of the location in which the preserved live register is stored. Register L immediately follows register A in DOCTOR.
X	has the value of the location in which the preserved index register is stored. Register X immediately follows register L in DOCTOR.
F	has the value of the register containing the lowest location being used by DOCTOR for symbols. Its contents will change from time to time, as symbols are defined. Register F immediately follows register X in DOCTOR.
M	has the value of the register which contains the <u>mask</u> used in searches (see below). Register M+1 contains the lower limit of all searches, and M+2 contains the upper limit. Register M immediately follows register F in DOCTOR.

<u>CHARACTER</u>	<u>ACTION</u>
Q	has the value of the last quantity typed by DOCTOR or you.
1	causes the last three characters typed in to be taken as their flexo code value. This applies only to letters or numerals.
8	print integers in octal.
ō	print integers in decimal.
,	when preceded by a legal symbol, causes that symbol to be defined as the current location.
_	when preceded by a legal symbol, causes that symbol to be defined as the address part of the last quantity typed by DOCTOR or you.
K	deletes all but initial symbols by setting the contents of F back to its initial value. Any redefinitions of initial symbols are not affected.
:	sets the symbol definition value to the expression typed by either DOCTOR or the operator beforehand. (See below)
)	causes the legal symbol typed immediately preceding the) to be defined as the current symbol value, as set by : or ,
S	sets the mode in which DOCTOR types out words to <u>symbolic</u>.
C	sets the mode in which DOCTOR types out words to octal <u>constants</u>.
R	sets the mode in which DOCTOR types out locations to <u>relative</u> (symbolic).
O	sets the mode in which DOCTOR types out locations to <u>octal</u>.
W	causes DOCTOR to search memory between the limits specified in M+1 and M+2, for <u>words</u> equal to the expression

CHARACTER ACTION

preceding the W. Only bits masked 1 in register M are compared. All occurrences are typed out with their locations. Typing W alone is an error. DOCTOR will not search itself.

N same as W, but finds all words not equal to the expression typed preceding the N.

E causes DOCTOR to search memory for all words whose address is equal to that of the expression preceding the E.

delete deletes all typed input since last DOCTOR printout, unless the operator has typed an intervening carriage return.

case shifts inform DOCTOR of the case in which the operator is typing, are otherwise ignored.

B conditions DOCTOR to insert a breakpoint at the location specified before the B. If no such location was specified, DOCTOR removes the previous breakpoint. A breakpoint is actually inserted only when a G, P, or U is executed (see below). DOCTOR will remove the instruction at the break location, and will save it for future restoration. The instruction at the break-location is only executed after the proceed is given.

P after the break trap occurs, causes DOCTOR to proceed with the user's program. The proceed will cause the instruction which was at the break-location to be executed and control to return to the user's program at the point at which it was interrupted, after all registers and indicators have been restored. If the breakpoint was moved after a trap, control will still return to the instruction trapped by the last breakpoint.

CHARACTER ACTION

U execute the preceding expression as an instruction. The breakpoint, if any, and all registers and indicators will be set up and saved.

G go to the location specified before the G. All indicators and registers will be restored, and the breakpoint, if any, will be inserted. Typing G alone is an error.

Y read a binary tape in standard binary block format. The tape is read into storage between the limits specified in M+1 and M+2. If a checksum error is encountered, the program will stop. It is then possible to move the tape back one block, and press Restart to continue reading, if desired.

T read MIDAS symbol table, and merge it with the existing symbol table. Definitions on tape take precedence over definitions in storage. The new contents of register F are typed out upon completing reading the symbol section of the tape. Checksum errors are handled as in Y. A number preceding T is taken as relocation to be applied to relocatable symbols.

V verify: reads a binary tape in binary block format and compares it against memory between locations specified by M+1 and M+2. No change is made to memory. Discrepancies are typed out as:

```
        location/          memory    tape
```

Checksum errors are handled as in Y.

CHARACTER	ACTION
H	puts DOCTOR into the title punch listen loop. Characters typed in are punched out in readable format on paper tape. The terminating characters are tab, carriage return, or backspace, which do the following. tab: sets DOCTOR to punch read-in mode data blocks. car. ret.: punches a standard input routine, and sets DOCTOR to punch standard checksummed data blocks. backspace: sets DOCTOR to punch standard checksummed data blocks, but punches no input routine. (A trn 17756 will be punched instead.)
-	when a register is open, make the modification, if any, and punch a one word block containing that register, in format specified by H (see above).
fa:laD	punches data blocks from fa through la in format specified by H (above). fa and la are any symbolic expressions.
J	punch a start (jump) block to the address specified to denote end of binary tape.
Z	zero all memory between register 0 and the lowest register used by DOCTOR (contents of register F).
fa:laZ	zero memory between fa and la except that part, if any, occupied by DOCTOR.

HINTS AND KINKS

Breakpoints are extremely useful for investigating misbehavior of long programs. Do not try to break at program-modified instructions, or TSX's followed by program parameters to be picked up by subroutines.

If the operator types an undefined symbol, DOCTOR will respond with a U. All typed input up to that point is deleted automatically.

If when attempting to type out a word as flexo code, the typewriter should hang, pressing Start Read will clear it.

When trying to determine the best symbol to fit a given value, and given two equally good symbols, DOCTOR will pick the one last defined for its printout.

There are two ways to print a block of registers. Either set the mask to zero, set up M+1 and M+2 to enclose the area to be printed and search for any word; or, if irrelevant parts of memory happen to contain zero, merely do an N-search for zero. If you change the mask or search limits, it is well to set them back to their usual values when you are through.

Author R. A. Saunders   Approved J B Dennis
R. A. Saunders   J. B. Dennis

Summary of Control Characters

A	accumulator storage
B	insert breakpoint
C	print words as constants
D	punch data blocks
E	address search
F	lowest location in Doctor
G	go to
H	enter title punch (header) mode
I	equals as instruction
J	punch start block
K	kill defined symbols
L	live register storage
M	mask register
N	not word search
O	print addresses in octal
P	proceed
Q	last quantity
R	print locations in symbolic (relative)
S	print words in symbolic
T	read symbol table
U	execute as instruction
V	verify tape against memory
W	word search
X	index register storage
Y	read binary tape
Z	zero memory
0-9	numerals and symbol constituents
a-z	symbol constituents
1	take as flexo code
3	print as flexo code
8	print integers in octal
o	print integers in decimal
:	set first argument value
|	examine register
/	examine register, print in symbolic
(	examine register, print in octal
)	define symbol
=	equals as octal constant
.	current location or take as decimal
,	define symbol as current location
_	define symbol as address typed
-	minus
+	plus
‾	(upper case minus) punch this register
tape feed	modify and open previous register
delete	delete
tab	modify and open addressed register
bk sp	modify and open next register
car ret	modify and close register
uc,lc	set case
space	plus
all other	ignored, but respond with X